

# Evaluación experimental de la capacidad de IEEE 802.11b para soporte de VoIP\*

Guefry Leider Agredo Méndez

Miembro IEEE. Grupo I+D Nuevas Tecnologías en Telecomunicaciones

Jaime Andrés Gaviria Molano

Miembro IEEE. División de Tecnologías de Información y Comunicación  
Universidad del Cauca, Popayán – Colombia  
{gagredo,jgaviria}@unicauca.edu.co

Fecha de recepción: 30-05-06

Fecha de selección: 30-10-06

Fecha de aceptación: 30-08-06

## ABSTRACT

Wireless LAN have been widely deployed worldwide using Technologies based on 802.11b/g standards mainly. But, those networks do not provide any support for QoS demanding services because the most of them were deployed before the 802.11e standard. However, in many places the need for QoS demanding services, like Voice over IP, has aroused and it is important to establish how those services will behave in Non QoS supporting WLAN's. This papers shows experimental results for a testbed, intended to establish the capacity of a legacy 802.11 LAN to support VoIP services.

## KEYWORDS

MAC, DCF, QoS, VoIP, Wireless LAN, 802.11, ITG, Asterisk

## RESUMEN

A la fecha existe gran cantidad de redes inalámbricas de área local implementadas a nivel mundial sobre los estándares 802.11a/b/g pero que no proveen ningún tipo de control o acciones para ofrecer QoS puesto que son previas a la generación del actual 802.11e. Sin embargo, en varios de estos lugares se ha visto la necesidad de realizar la transmisión de VoIP sobre estas redes y es muy importante conocer los resultados que se obtienen al evaluarlo de forma experimental

\* Manuscrito presentado ante i2Comm 2006. Este trabajo fue financiado en parte por Unicauca en el marco de desarrollo de la Maestría en Ingeniería.

teniendo en cuenta una gran cantidad de factores que no se pueden tener en escenarios de simulación. En este artículo se presentan los resultados de evaluar de forma experimental la capacidad de 802.11b para soportar comunicaciones de VoIP; para tal efecto se realizó la verificación de dos formas, usando el generador de tráfico ITG y haciendo la generación de llamadas reales usando la PBX-VoIP Asterisk. Como resultado, se encontró que lo obtenido por ambas vías fue en general coincidente y que en las redes inalámbricas sin ningún tipo de manejo de QoS el principal limitante en la capacidad y en el número de llamadas concurrentes es la contienda por el acceso al medio y no tanto el tipo de códec usado y/o el ancho de banda que consume cada llamada. Al final se pueden observar los resultados resumidos en tablas, que

muestran la máxima cantidad de llamadas sin degradación de la calidad para cada códec usado. Se concluye también que en una red 802.11b la capacidad de canales de VoIP no puede calcularse ni aproximarse con una simple división entre ancho de banda total y ancho de banda por canal, sino que deben considerarse más factores, pues, lo que se observa inicialmente es una degradación cuasi-exponencial al aumentar el número de comunicaciones hasta cierto punto y luego se tiene una degradación abrupta que incluso hace caer las demás comunicaciones de voz que se estén realizando en ese momento.

#### **PALABRAS CLAVE**

MAC, DCF, QoS, VoIP, Wireless LAN, 802.11, códecs, ITG, Asterisk

**Clasificación Colciencias: A**

## I. INTRODUCCIÓN

Las redes inalámbricas IEEE 802.11[1], conocidas también con el nombre comercial de *Wi-Fi*[2], se han convertido en las redes de datos inalámbricas con mayor tasa de penetración a todos los niveles: público (*Hotspots*), empresarial, SoHo y en el hogar; inclusive y como aspecto relevante esta tecnología ha hecho viable la comunicación de datos y voz a zonas rurales [3] y el establecimiento de comunidades inalámbricas. Esta tendencia ha despertado el interés en la fabricación de dispositivos para redes inalámbricas, tales como televisores, DVD, consolas de videojuegos y otros. De esta forma se hace evidente la necesidad de contar con esquemas de calidad de servicio (QoS) que permitan que flujos de audio y video generados desde un DVD o computador puedan ser fácilmente distribuidos a televisores, equipos de sonido o parlantes en cualquier lugar del hogar.

Así, la tecnología de las redes IEEE 802.11 se convierte en la tecnología inalámbrica de preferencia para el transporte de estos tipos de información debido principalmente a dos factores claves: sus relativas altas tasas de transferencia de datos y su carácter dominante en el mercado.

Avances recientes en la familia IEEE 802.11, especialmente la finalización de la norma IEEE 802.11e, han generado el marco propicio para la consolidación del rol de 802.11 en las aplicaciones de Voz sobre IP. 802.11e; es una extensión a los estándares 802.11a/b/g para proporcionar soporte de calidad de servicio (QoS) a aplicaciones de tiempo real o de contenido sensible a retardo tales como voz y video. La norma IEEE 802.11e

introduce la función de coordinación híbrida (HCF, *Hybrid Coordination Function*) como el esquema de control de acceso al medio. Esta especificación es compatible con los sistemas tradicionales de las redes 802.11, esto es: la función de coordinación distribuida (DCF, *Distributed Coordination Function*) y la función de coordinación puntual (PCF, *Point Coordination Function*)

A pesar de lo anterior, la norma IEEE 802.11e no se encuentra disponible en las instalaciones actuales de redes inalámbricas ni en los dispositivos móviles idóneos para la transmisión de VoWLAN como PocketPC y PALMS, por lo que resulta interesante evaluar la capacidad de estos sistemas, para soportar comunicaciones de voz con diferentes códecs usando redes inalámbricas 802.11b.

Un paquete típico de VoIP consta de 40 Bytes conformados por los encabezados IP/UDP/RTP y una carga útil de entre 10 y 30 bytes, lo que depende del códec que se utilice. Por tanto, la eficiencia en el nivel IP para VoIP es menos del 50%. En el nivel físico y en el de enlace de 802.11, la pérdida de eficiencia aumenta, así por ejemplo, para el códec de GSM, con una carga útil de 33 bytes, el tiempo de transmisión a 11 Mbps es  $33 \times 8 / 11 = 24 \mu s$  y el tiempo de transmisión para el encabezado IP/UDP/RTP de 40 bytes es  $40 \times 8 / 54 = 29 \mu s$ . Sin embargo, estos niveles tienen una carga adicional de más de 800  $\mu s$ , atribuida al preámbulo de nivel físico, al encabezado MAC, al tiempo de contención, a la espera de ACK y al intercambio entre transmisión y recepción. Como resultado, la eficiencia general cae aproximadamente al 30%.

En una WLAN empresarial o en *Hotspots* públicos, el soporte de VoIP se torna aún más complicado, dado que se requiere el soporte simultáneo de otras aplicaciones aparte de VoIP. La necesidad de atender este tipo de escenarios limita el número de sesiones de VoIP posibles.

## II. CALIDAD DE SERVICIO

“Calidad de Servicio (QoS, *Quality of Service*) se refiere a la capacidad de la red para proporcionar un mejor servicio a tráfico seleccionado de varias tecnologías de red. QoS procura que el tráfico de tiempo real de aplicaciones multimedia y de voz reciba la más alta prioridad, el mayor ancho de banda y el menor retardo en comparación con el tráfico de datos considerado como de ‘al mejor esfuerzo’”. [4]

Las tecnologías de calidad de servicio proporcionan la base para el éxito de las aplicaciones multimedia y de voz y para el contexto de este trabajo especialmente en el escenario inalámbrico. Esta calidad está determinada por factores como:

- Retardo (Latencia): cantidad de tiempo que le toma a un paquete alcanzar al receptor luego de ser enviado por el transmisor. También se conoce como retardo de extremo a extremo.
- *Jitter*: diferencia en el retardo de extremo a extremo entre varios paquetes.
- Pérdida de paquetes: medida comparativa de los paquetes transmitidos y recibidos exitosamente con respecto al total de paquetes transmitidos.

Uno de los roles de la calidad de servicio es mantener el retardo, el *jitter*

y las pérdidas de paquetes para los tipos de tráfico seleccionados dentro de límites aceptables. Los requerimientos que se deben cumplir cuando se trabaja con voz sobre IP son:

- Retardo máximo en un sentido no mayor de 150 ms (de acuerdo con la recomendación ITU-T G.114). Sin embargo, es importante considerar que los usuarios normalmente notarán los retardos de la voz si estos en viaje redondo sobrepasan los 250 ms [5].
- Pérdida de paquetes mínima: VoIP no es tolerante a las pérdidas de paquetes, aun con un 1% de paquetes perdidos se puede degradar enormemente una comunicación de voz así se esté utilizando el códec G.711, en caso de utilizar códecs con mayor tasa de compresión la pérdida es prácticamente intolerable [5] [6].
- El *jitter* promedio no debe ser mayor que 30 ms [5], aunque algunos autores hablan de hasta 50 ms [7][8].

Partiendo de lo anterior, para establecer la capacidad de la red inalámbrica para soportar llamadas de VoIP se tomaron los límites anteriores, y si al realizar una llamada adicional se sobrepasaba la pérdida de paquetes, el retardo o el *jitter*, se consideraba que se había logrado establecer la capacidad para cada códec.

## III. MECANISMOS DE ACCESO EN 802.11

802.11 trabaja dos mecanismos de acceso al medio, uno obligatorio conocido como DCF (Distributed Coordination Function) y otro opcional conocido como PCF (Point Coordina-

tion Function). El segundo método rara vez es implementado en equipos del mercado, y tampoco está presente en los equipos usados en este laboratorio. Por este motivo sólo se cuenta con el esquema DCF que se describe a continuación.

DCF trabaja en un esquema “revisar-antes\_de-enviar”, basado en CSMA (Carrier Senses Multiple Access). Cualquier estación primero detecta si existe alguna transmisión en curso en el medio inalámbrico (revisa), y solamente cuando lo encuentra libre puede transmitir (envía). Sin embargo, si dos estaciones detectan libre el medio al mismo tiempo, podría ocurrir una colisión. Por esto, 802.11 define un mecanismo para “evitar la colisión”, por medio del cual todos deben esperar un tiempo aleatorio antes de enviar para evitar colisiones, de esta forma cada estación debe desarrollar un procedimiento de *backoff* antes de iniciar, así, cada estación debe permanecer escuchando el medio por un intervalo aleatorio con una duración mínima de un DIFS (DCF Inter Frame Space) luego que ha detectado el medio libre.

El tiempo de *backoff* se escoge en el intervalo (0, CW-1) conocido como la ventana de contención (*Contention Window*); ésta consta de dos parámetros CWmin y CWmax. Inicialmente el número aleatorio está entre 0 y CWmin. Si el *backoff* original expira antes de que pueda enviarse la trama, el contador se aumenta y el tamaño de la ventana de contención se dobla. Esta situación continúa hasta que se alcance el CWmax. Los reintentos continúan hasta que se venza el tiempo de vida de los paquetes (TTL). Luego de cada transmisión exitosa

de una trama, la estación receptora devuelve un reconocimiento (ACK) inmediatamente. Si luego del envío de una trama no se recibe el ACK se asume que la transmisión no ha sido exitosa.

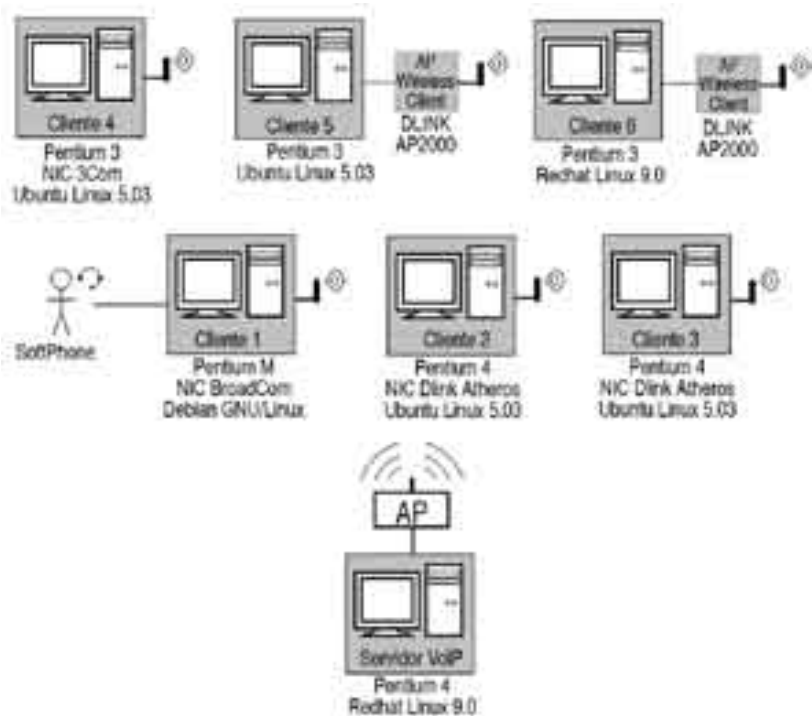
#### IV. MONTAJE DEL SISTEMA

Para el montaje del sistema y ante la necesidad de aproximarse a un entorno de operación real, se realizó el montaje tal como se indica en la Figura 1, en donde se contó con seis clientes conectados inalámbricamente a un punto de acceso y éste a su vez conectado por su interfaz ethernet a 100Mbps con el servidor de VoIP. Se trabajó en un ambiente heterogéneo en cuanto a modelos y marcas de equipos que trabajan con 802.11b, así mismo los computadores usados tuvieron diferentes características hardware y ubicación en el laboratorio. El sistema operativo usado fue GNU/Linux, en dos distribuciones: Ubuntu y RedHat. En las Figuras 2 y 3 se muestran fotografías de la infraestructura utilizada.

La descripción de la infraestructura se resume en la Tabla I.

Como se aprecia en la Figura 1, el servidor de VoIP se encuentra conectado por medio cableado al punto de acceso y no conectado inalámbricamente, pues es el caso típico de una implementación real, ya que es poco frecuente que dos estaciones asociadas al mismo punto de acceso requieran una comunicación de voz entre ellas. Las llamadas generalmente van hacia otra red, o a la misma red pero a un equipo asociado a otro punto de acceso.

En cuanto a los dispositivos inalámbricos, se contó con equipos 802.11b y



**Figura 1.** Laboratorio montado.



**Figura 2.** Infraestructura de laboratorio utilizada.



**Figura 3.** Equipo cliente usando Softphone, Asterisk e Iptraf.

**Tabla I.** Descripción de la infraestructura utilizada.

Equipo/ Características	CPU	Memo- ria	Red	Sistema operativo	Software	Dirección IP
Ciente1	Pentium M - 1.5Ghz	512MB	BroadCom Wireless	Debian GNU/Linux 3.1	Asterisk PBX 1.0.9 - ITG	192.16 8.1.100
Ciente2	Pentium 4 - 2.6Ghz	512MB	Dlink / Atheros Wireless	Ubuntu GNU/Linux 5.03	Asterisk PBX 1.0.6 - ITG	192.16 8.1.101
Ciente3	Pentium 4 - 2.6Ghz	512MB	Dlink / Atheros Wireless	Ubuntu GNU/Linux 5.03	Asterisk PBX 1.0.6 - ITG	192.16 8.1.102
Ciente4	Pentium 3 - 800Mhz	128MB	3COM PCMCIA Wireless	Ubuntu GNU/Linux 5.03	Asterisk PBX 1.0.6 - ITG	192.16 8.1.103
Ciente5	Pentium 3 - 800Mhz	128MB	DLINK AP2100 Wireless Client Mode	Linux RedHat 9.0	Asterisk PBX 1.0.7 - ITG	192.16 8.1.104
Ciente6	Pentium 3 - 500Mhz	128MB	DLINK AP2100 Wireless Client Mode	Ubuntu GNU/Linux 5.03	Asterisk PBX 1.0.6 - ITG	192.16 8.1.105
Servidor VoIP	Pentium 4 - 2.6Ghz	512MB	Atheros Wireless	Ubuntu GNU/Linux 5.03	Asterisk PBX 1.0.6 - ITG	192.16 8.1.2
Punto de Acceso Linksys WRT54G	BCM330 2 - 216 Mhz	16 MB	BCM330 2	Linux OpenWRT	wireless utils	192.16 8.1.1

802.11g pero se forzó la configuración de los equipos 802.11g para que trabajaran exclusivamente con la norma 802.11b. La conexión y ubicación de los equipos fue tal que todos trabajarán a 11Mbps durante el tiempo de las pruebas, sin posibilidad de que se negociara a una velocidad menor. Los dispositivos usados fueron tarjetas inalámbricas PCI, PCMCIA, miniPCI e incluso puntos de acceso configurados como *wireless client*, obteniendo un ambiente totalmente heterogéneo con diferentes marcas y tipos de equipos. En cuanto al sistema operativo, se instaló GNU/Linux en todos los equipos, tanto cliente como servidor, e incluso en uno de los puntos de acceso, lo que dio una flexibilidad enorme en configuraciones, toma de datos y permitió el uso de una gran cantidad de herramientas como Iptraf para

el diagnóstico de interfaces de red, ITG para la generación de Tráfico de Internet y Asterisk como Servidor de VoIP, los cuales se detallarán en la siguiente sección

## V. METODOS DE EVALUACIÓN

Para realizar el trabajo experimental y establecer la capacidad se recurrió al uso de dos herramientas “open source” encargadas de generar las llamadas. Una fue el Generador de Tráfico de Internet – ITG [13] que tiene la capacidad de generar tráfico de VoIP con tres códecs: G.711, G.729 y G.723, la otra fue el Servidor de VoIP Asterisk® [10] con capacidad para soportar los mismo códecs y algunos más, sin embargo para el ejercicio comparativo se tomaron los mismos del ITG. Para verificar la ocupación del canal (ancho de banda)

y cantidad de paquetes por segundo, se probó con una llamada en ambas herramientas. Los resultados mostraron que para G.711 y G.729 las dos formas de generar llamadas eran equivalentes, pero para G.723 hubo cambios que claramente se dedujeron al revisar la diferencia en cuanto a características de este códec en ITG y en Asterisk, como se pueden apreciar en la Tabla II para ITG y en la Tabla III para Asterisk.

Tabla II. Atributos de los códecs de ITG.

Códec	G.711	G.723.1	G.729
Tasa de Bits (Kbps)	84	16.6	28
Paquetes /segundo	50	26	50.

Tabla III. Atributos de los códecs de Asterisk.

Códec	G.711	G.723.1	G.729
Tasa de Bits (Kbps)	84	18.2	28
Paquetes /segundo	50	26	50

A. Evaluación de la capacidad con ITG

Para realizar el proceso de evaluación por este camino, se descargó el paquete D-ITG, Distributed Internet Traffic Generator Versión 2.4 de la URL

<http://www.grid.unina.it/software/ITG/download.php>.

Luego se copió a cada uno de los equipos clientes en sus respectivos directorios y también en el equipo central, tal como lo muestra el siguiente comando:

```
# scp D-ITG-2.4.zip
root@192.168.1.104:/root
```

Para el proceso de instalación se deben seguir estos pasos: Una vez se copia, se debe descomprimir

\$unzip D-ITG-2.4.zip lo cual crea directorios y descomprime archivos. Con lo anterior aparece un nuevo directorio que es el ITG y como se tiene el código fuente, éste debe compilarse para lo cual hay que cambiarse a ITG/src

```
invitado@ryst15:~$ cd ITG/
src/
invitado@ryst15:~/ITG/src$
```

Estando en el directorio se ejecuta el comando make que requiere tener instalado el compilador g++. En caso de no tenerlo aparece este mensaje: invitado@ryst15:~/ITG/src\$ make

```
g++ -DLINUX_OS -Wall -Wno-deprecated -c -o common/thread.o common/thread.cpp make: g++:
No se encontró el programa
make: *** [common/thread.o]
Error 127
```

Si esta es la situación debe instalarse, por ejemplo para el caso de ubuntu-linux que fue el sistema operativo utilizado, estando como superusuario se ejecuta apt-get install g++, es necesario tener el CD de instalación a mano, pues es solicitado.

```
root@ryst15:~/ITG/src$ apt-get install g++
```

Si es con Redhat se hace por medio de la instalación del RPM apropiado.

Luego de tener instalado este paquete ya se puede compilar el D-ITG con el comando antedicho.



```
[root@ryst7 bin]# cd src
```

```
[root@ryst7 src]# make
```

Con esto aparecen los binarios en el directorio ITG/bin. Para las estaciones clientes se trabaja el programa ITGSend y para el equipo central ITGRecv. El ITG permite dos formas de calcular el retardo de un paquete, si es en una vía se requiere un servidor NTP para sincronización de relojes y el receptor –ITGRecv– es el encargado de tomar los datos, si es el retardo de viaje redondo no se requiere el servidor NTP porque el transmisor –ITGSend– es el encargado de obtener los datos. Inicialmente se trató con NTP pero la sincronización se perdía rápidamente, además se consideró que la llamada sería en doble vía por lo que también era conveniente la segunda forma, por este motivo la toma de datos se tuvo en cada una de las seis estaciones generadoras, cada una de las cuales generó un archivo (log) que al pasarlo al ITGDec se decodifica y presenta en formato *humanreadable*. En este se encuentran los datos de retardo, *jitter* y pérdida de paquetes.

La condición límite se estableció basándose en los parámetros definidos en la Sección II, de tal forma que la comunicación es inviable cuando se supere cualquiera de los límites de pérdidas, *jitter* o retardo mencionados.

En cada una de las estaciones clientes se construyó un script para que el ITGSend generara las llamadas que contenían las tramas de VoIP enviadas al equipo central, con los parámetros requeridos tales como cantidad y códec.

La sintaxis fue:

```
-a 192.168.1.2 -rp 10001 -t 60000  
VoIP -x G.711.2 -h RTP
```

Como se trabajó con 6 PC los puertos se escogieron colocando el primer número de acuerdo con el cuarto byte de la dirección IP tal como aparece a continuación:

Cliente 1 - 192.168.1.101 -rp 10001-1000xx

Cliente 2 - 192.168.1.102 -rp 20001-2000xx

Cliente 3 - 192.168.1.103 -rp 30001-3000xx

Cliente 4 - 192.168.1.104 -rp 40001-4000xx

Cliente 5 - 192.168.1.105 -rp 50001-5000xx

Cliente 6 - 192.168.1.106 -rp 60001-6000xx

Se comenzó en forma secuencial con una conexión de VoIP con el Cliente 1, luego una segunda con el Cliente 2 y así sucesivamente hasta llegar al Cliente 6, de esta manera seis llamadas activas significaban que los seis clientes estaban generando llamadas. Como no se disponía de más clientes inalámbricos para que hubiera más llamadas se aumentó a que cada cliente generara dos o más llamadas, de tal forma que 12 llamadas implican que cada uno de los seis está generando dos y que 18 están generando tres.

El archivo de registro (log) que se obtuvo con cada uno se pasó al ITGDec para que lo decodificara y de esta forma conocer los datos de retardo, *jitter* y pérdida de paquetes. El límite se estableció por superar cualquiera de ellos, pero como se mencionó anteriormente fue interesante ver que cuando se sobrepasaba la capacidad, todos estos límites eran considerablemente rebasados.

Se realizó primero con G.711, luego con G.729 y finalmente con G.723.

En el equipo central se ejecutó la aplicación Iptraf para establecer ancho de banda utilizado y paquetes por segundo tomando *screenshots* que comprobaran la utilización del canal y revisando si la entrada y/o salida era  $n \times [BW_c]$  donde  $n$  era la cantidad de conexiones y  $BW_c$  el ancho de banda utilizado por cada códec. Para G.711  $BW_c=64$ , para G.729  $BW_c=29$  y para G.723  $BW_c=26.6$ . Siempre se utilizó un tipo de protocolo RTP y no se utilizó VAD (Voice Activity Detection).

Antes de iniciar las pruebas se envió un archivo de 1.9 GB a cada cliente para ver si existía un promedio de velocidad relativo entre todos, con el fin de descartar problemas de desempeño en alguno de ellos, las velocidades relativamente fueron cercanas, razón por la cual se consideró que todos se podían utilizar.

#### B. Evaluación de la capacidad con asterisk

Después de haber trabajado con el generador de tráfico, se decidió realizar el laboratorio con un servidor de VoIP generando llamadas reales entre los equipos y poder hacer una comparación y validación de resultados respecto al generador de tráfico D-ITG. Para llevar a cabo esta evaluación fue necesario instalar la PBX IP Asterisk, realizar la configuración de los clientes y escribir algunos scripts que permitieran la generación automática de llamadas desde los clientes hacia el servidor, la respuesta de las llamadas en el servidor, y la reproducción automática de lado y lado de mensajes de voz pregrabados, de modo que se obtuvieran datos totalmente reales en una conversación de VoIP. Se tuvo el problema que si

se usaban *softphones* como clientes, se tenía la limitación de tan solo una llamada por cliente incluso en los *softphones* que manejan llamadas simultáneas, pues estos realmente solo manejan una comunicación al tiempo y las demás llamadas las pone en espera, además de que cada llamada por *softphone* no se podría programar, ni ponerle a reproducir un mensaje automáticamente. Para solucionar este inconveniente se instaló el servidor de VoIP asterisk en todos los clientes, ya que asterisk también puede operar como cliente, de modo que todo el trabajo, configuración y escritura de scripts se hizo para manejar las llamadas en asterisk. Adicionalmente al realizar la evaluación, se buscó tener una verificación “audible” al ser humano de la calidad de la voz, que demostrara cómo el aumento en el  *jitter* y retardo realmente afecta la percepción y la calidad de la comunicación. Para tal efecto, se instaló adicionalmente en uno de los clientes un *softphone*, con el cual se llamó al servidor cada vez que se quiso evaluar la calidad. Para más información sobre asterisk y su configuración se puede visitar la URL [www.asterisk.org](http://www.asterisk.org).

A continuación se explican sólo las partes claves de la configuración, para facilitar la reproducción del experimento.

## I. CONFIGURACIÓN DEL SERVIDOR

### 1.1. Sección de clientes SIP “sip.conf”

En esta sección se declaró el abonado sip que se utilizó para monitorear la calidad de audio a través de parlantes

```

; Se declara la sección general;
Se declara la sección de abonado sip con un nombre
; cualquiera, para este caso [softphone1]
[softphone1]
type=friend          ;Permite hacer y recibir llamadas
context=ciclo        ;Podrá llamar a los números que se ;in-
                      ;cluyan en el contexto ciclo del
                      ;plan de marcado
host=dynamic          ;podra iniciar sesión desde cualquier
                      ;equipo
secret=telefono1      ;La contraseña
qualify=yes           ;monitorea la conexión y el retardo
dtmfmode=rfc2833      ;Detección de tonos estandar
relaxdtmf=yes         ;Facilita la detección de tonos

```

## 1.2. Sección de plan de marcado “extensions.conf”

En esta sección se declara la lógica y el flujo de la llamada

```

;Se declara la sección general
;Se declara el contexto que maneja las llamadas
[ciclo]          ;Contexto Ciclo
exten=s,1,BackGround,demo-instruct
;Reproduce el mensaje demo-instruct que esta ya grabado
;en el servidor
exten=s,2,Goto(ciclo,s,1)
;Crea un ciclo para que se repita el mensaje
;indefinidamente
exten=101,1,Goto(ciclo,s,1)
;cuando el softphone llama a 101 entra al ciclo

```

### 1.3. Sección de clientes IAX “iax.conf”

```
;Se declara la sección general
;Se declara uno por uno los clientes a los que se
;conectará
[cliente1]
type=friend          ;Puede hacer y recibir llamadas
host=192.168.1.100   ;Se conectará desde esa IP
context=ciclo        ;Podrá llamar a los números que se
                    ;declaren en ese contexto
qualify=yes          ;Se monitorea la conexión
                    ;y el retardo
disallow=all         ;No permite un códec diferente a
allow=ulaw           ;g711u
[cliente2]
type=friend          ;Puede hacer y recibir llamadas
host=192.168.1.101   ;Se conectará desde esa IP
context=ciclo        ;Podrá llamar a los números que se
                    ;declaren en ese contexto
qualify=yes          ;Se monitorea la conexión y el retar-
                    ;do
disallow=all         ;No permite un códec diferente a
allow=ulaw           ;g711u
;Se continúa con los demás clientes
```

Para cambiar de códec se puede modificar el allow por:

alaw    Códec G.711a  
ulaw    Códec G.711u  
g729    Códec G.729  
g723    Códec G.723  
gsm     Códec GSM  
ilbc    Códec ilbc

Es de anotar que los códecos g729 y g723 no son libres y en ciertos

ambientes debe pagarse por ellos, asterisk los incluye solo en modo *passthrough*, de modo que para poder generar las llamadas y recibirlas, se debió compilar los códecos gratuitos para uso académico disponibles en la página de intel [www.intel.com](http://www.intel.com) e incluirlos en los módulos de asterisk.

## 2. CONFIGURACIÓN DEL CLIENTE

En los clientes solo fue necesario realizar una configuración en los

clientes IAX, pues todas las llamadas solo se debe declarar el servidor al se harían manejando este protocolo, que irán conectados.

Sección IAX iax.conf

```
;Se declara la sección general
;Se declara el servidor que le llamaremos ap
[ap]
type=friend
host=192.168.1.2
context=ciclo
qualify=yes
disallow=all
allow=ulaw
trunk=no          ;Para que cada llamada se haga como
                  ;una
                  ;llamada independiente. Si se coloca
                  ;trunk=yes, se meten varias llamadas
                  ;por una
                  ;"conexión" que ahorra ancho de banda
                  ;pero
                  ;no es el caso real.
```

Una vez se haya iniciado asterisk en los clientes como en el servidor, y se haya hecho la configuración correcta, en la interfaz de línea de comandos CLI de asterisk, se pueden observar las conexiones de los clientes tal como aparece en la Figura 4.

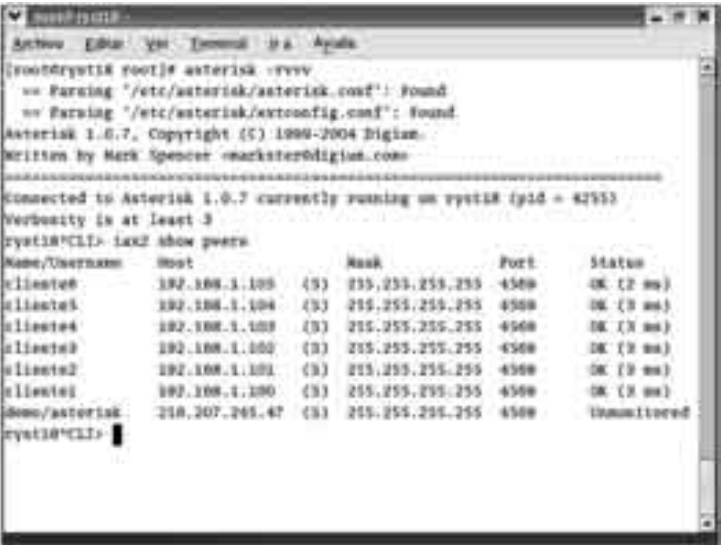


Figura 4. Estado de conexión en el servidor.

### 3. GENERACIÓN DE LLAMADAS

Una vez los clientes se encuentran conectados con el servidor, es necesario que se inicien las llamadas.

Asterisk chequea constantemente un directorio en donde se pueden colocar scripts de llamadas para que él ejecute inmediatamente.

El script se detalla a continuación.

```
;Script para generar llamadas debe ubicarse en /
;var/spool/asterisk/outgoing, tan pronto se copie el
script
;en este directorio, se hará la ejecución.
;Archivo sample.call
Channel: IAX2/ap ; llama al servidor 'ap'
MaxRetries: 2 ; hasta dos reintentos
RetryTime: 60 ;Reintento cada 60 segundos
WaitTime: 30 ;Esperara 30 segundos la respuesta
Context: ciclo ;La llamada llegará al contexto ciclo,
Extension: s ;extension s y
Priority: 1 ;prioridad 1
```

Si se observa en las configuraciones anteriores, este script lleva la llamada a la ejecución del archivo pregrabado, que se ejecuta tanto en el servidor como en los clientes.

Para generar las llamadas se puede hacer un script que copie el archivo anterior las veces necesarias con diferente nombre cada vez, así:

```
#!/bin/bash
#mpstat -P 0 1 1 adiciona un retardo de un segundo entre
#llamada y llamada generada además de mostrar el consumo
#de CPU
cp/tmp/sample.call /var/spool/asterisk/outgoing/sample1.
call
mpstat -P 0 1 1
cp/tmp/sample.call /var/spool/asterisk/outgoing/sample2.
call
mpstat -P 0 1 1
cp/tmp/sample.call /var/spool/asterisk/outgoing/sample3.
call
mpstat -P 0 1 1
```

Es de anotar también que este script debe ejecutarse en cada cliente, pues es cada cliente quien genera la(s)

llamada(s), en cuanto al servidor también es importante aclarar que para generar una llamada con diferentes características como cambio de códec, debe modificarse el archivo `iax.conf` tanto en los clientes como en el servidor cambiando `allow=ulaw` por otro códec.

[illegible]

**Figura 5.** Estado de cada llamada desde cada uno de los seis clientes.

En la Figura 6 se muestra el estado de seis llamadas generadas hacia el

servidor, una desde cada cliente

[illegible]

Figura 6. Generación de llamadas desde un cliente.

### A. Resultados obtenidos del trabajo con ITG

En primer lugar se tomó el dato de utilización de ancho de banda y canti-

dad de paquetes por segundo de cada códec por medio de IPtraf. Los resultados coincidieron exactamente con los que muestra la Tabla II, como se puede verificar en las Figuras 7-9.



Figura 7. Datos de Iptraf de una llamada con G.711.



Figura 8. Datos de Iptraf de una llamada con G.729.

Luego de esto se inició la generación de llamadas en la forma explicada anteriormente; con G.711 se lograron realizar hasta 11 llamadas, con G.729 hasta 14 y con G.723 hasta 28.

Luego se procedió a decodificar los archivos de registro (Logs) obtenidos con cada uno de los códecs: G.711, G.729 y G.723.





**Figura 9.** Datos de Iptraf de una llamada con G.723.

Para esto se utiliza el Decodificador del ITG, la sintaxis es:

```
./ITGDec [log]
```

En este caso los *logs* siguieron la siguiente convención de nombrado: Log-[cantidad de llamadas]-[códec].log. Por ejemplo, para el caso de la decodificación de 11 llamadas realizadas con G.711 este comando quedaría:

```
./ITGDec Log-11-G711.log
```

Los siguientes pares de figuras muestran los resultados obtenidos sobre la base de los cuales se estableció el límite de capacidad de acuerdo con cada códec. En la figura que aparece primero las condiciones de retardo *jitter* y pérdida de paquetes no superan los límites para una óptima comunicación de VoIP pero en la segunda (que es cuando se aumenta una llamada más) los supera considerablemente. En la parte inferior

de cada una aparece el comando utilizado para la decodificación de los resultados.

Estos resultados son los que dan soporte a las capacidades encontradas, esto es: para G.711 hasta 11 llamadas, para G.729 hasta 14 llamadas y para G.723 hasta 28 llamadas, lo anterior en razón de que siempre que se aumentaba *una sola* llamada, los límites de retardo (retardo) y pérdida de paquetes (packets dropped) eran sobrepasados “notoriamente”, lo cual se aprecia en las Figuras 11, 13 y 15 para G.711, G.729 y G.723 respectivamente, especialmente para la pérdida de paquetes (*Packets dropped*).

#### B. Resultados obtenidos del trabajo con asterisk

Inicialmente se tomaron los datos de consumo de ancho de banda para una sola llamada con cada uno de los tres códecs, obteniendo resultados similares a los obtenidos con ITG:

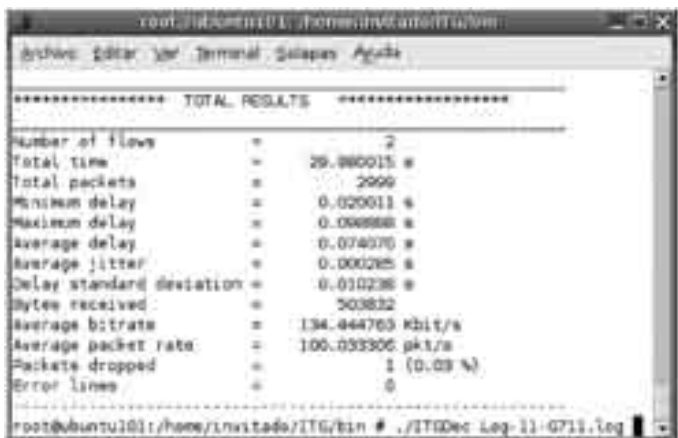


Figura 10. Decodificación de 11 llamadas con G.711.

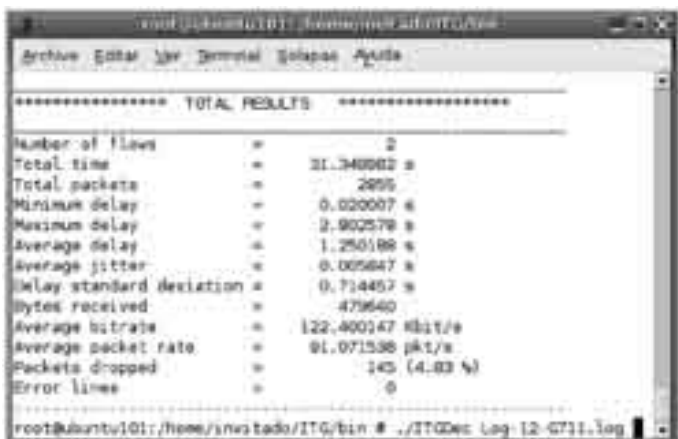


Figura 11. Decodificación de 12 llamadas con G.711.

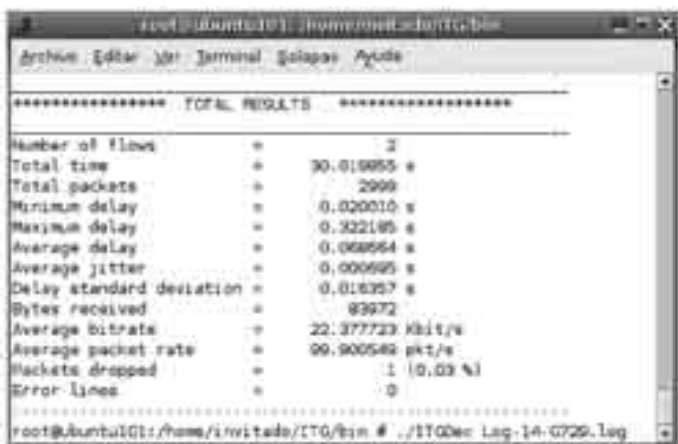


Figura 12. Decodificación de 14 llamadas con G.729.

```

root@ubuntu101: /home/invitado/ITG/bin-
Archive Editor for Terminal Sessions Ayuda
===== TOTAL RESULTS =====
Number of flows      =      3
Total time           =    30.425446 s
Total packets        =     4271
Minimum delay        =    0.020006 s
Maximum delay        =    2.245148 s
Average delay        =    1.014241 s
Average jitter       =    0.004959 s
Delay standard deviation =    0.551907 s
Bytes received       =    119588
Average bitrate      =   31.443173 Kbit/s
Average packet rate  =   140.371307 pkt/s
Packets dropped      =     229 (5.38 %)
Error lines          =      0
=====
root@ubuntu101: /home/invitado/ITG/bin # ./ITGDec Log-15-0729.log

```

Figura 13. Decodificación de 15 llamadas con G.729.

```

root@ubuntu101: /home/invitado/ITG/bin-
Archive Editor for Terminal Sessions Ayuda
===== TOTAL RESULTS =====
Number of flows      =      5
Total time           =    30.062761 s
Total packets        =     5879
Minimum delay        =    0.075936 s
Maximum delay        =    3.426430 s
Average delay        =    0.904287 s
Average jitter       =    0.007613 s
Delay standard deviation =    0.492350 s
Bytes received       =    147402
Average bitrate      =   99.225140 Kbit/s
Average packet rate  =   129.030065 pkt/s
Packets dropped      =      21 (0.34 %)
Error lines          =      0
=====
root@ubuntu101: /home/invitado/ITG/bin # ./ITGDec Log-28-0723.log

```

Figura 14. Decodificación de 28 llamadas con G.723.

```

root@ubuntu101: /home/invitado/ITG/bin-
Archive Editor for Terminal Sessions Ayuda
===== TOTAL RESULTS =====
Number of flows      =      5
Total time           =    30.187249 s
Total packets        =     3720
Minimum delay        =    0.838467 s
Maximum delay        =    1.852169 s
Average delay        =    1.210702 s
Average jitter       =    0.007074 s
Delay standard deviation =    0.496373 s
Bytes received       =    141360
Average bitrate      =   37.561719 Kbit/s
Average packet rate  =   123.558267 pkt/s
Packets dropped      =     180 (4.82 %)
Error lines          =      0
=====
root@ubuntu101: /home/invitado/ITG/bin #

```

Figura 15. Decodificación de 29 llamadas con G.723.

- G.711 82.4 Kbits/sg
- G.729 27.8Kbits/sg
- G.723 18.3Kbits/sg

Es importante anotar también que se tomaron medidas para varias llamadas y el consumo de ancho de banda fue exactamente el consumo

de una sola llamada multiplicado por el número de llamadas, tal como se observa en las Figuras 16 y 17; además, se consumió el mismo ancho de banda cuando se hicieron llamadas desde diferentes clientes que cuando se hicieron llamadas desde un solo cliente.



Figura 16. Datos de Iptraf con una llamada G.711.



Figura 17. Datos de Iptraf con diez llamadas G.711.

Para hacer una evaluación de los resultados obtenidos se tomaron datos empezando por una llamada y terminando en el momento en que la cantidad de llamadas no permitieran una comunicación fluida.

## I. RESULTADOS CON G.711

El tipo de medidas que se tomaron se pueden observar en la Figura 18 que muestra los resultados tabulados a medida que se iban generando nuevas llamadas.

Callname	From	To	ISDN	Seq	DTM	Seq	Jitter	Delay	Codec
00000000000000000000	100.100.1.100	100.100.1.100	00000000000000000000	00000000000000000000	00000000000000000000	00000000000000000000	00000000000000000000	00000000000000000000	G.711
00000000000000000000	100.100.1.100	100.100.1.100	00000000000000000000	00000000000000000000	00000000000000000000	00000000000000000000	00000000000000000000	00000000000000000000	G.711
00000000000000000000	100.100.1.100	100.100.1.100	00000000000000000000	00000000000000000000	00000000000000000000	00000000000000000000	00000000000000000000	00000000000000000000	G.711
00000000000000000000	100.100.1.100	100.100.1.100	00000000000000000000	00000000000000000000	00000000000000000000	00000000000000000000	00000000000000000000	00000000000000000000	G.711
00000000000000000000	100.100.1.100	100.100.1.100	00000000000000000000	00000000000000000000	00000000000000000000	00000000000000000000	00000000000000000000	00000000000000000000	G.711
00000000000000000000	100.100.1.100	100.100.1.100	00000000000000000000	00000000000000000000	00000000000000000000	00000000000000000000	00000000000000000000	00000000000000000000	G.711
00000000000000000000	100.100.1.100	100.100.1.100	00000000000000000000	00000000000000000000	00000000000000000000	00000000000000000000	00000000000000000000	00000000000000000000	G.711
00000000000000000000	100.100.1.100	100.100.1.100	00000000000000000000	00000000000000000000	00000000000000000000	00000000000000000000	00000000000000000000	00000000000000000000	G.711
00000000000000000000	100.100.1.100	100.100.1.100	00000000000000000000	00000000000000000000	00000000000000000000	00000000000000000000	00000000000000000000	00000000000000000000	G.711
00000000000000000000	100.100.1.100	100.100.1.100	00000000000000000000	00000000000000000000	00000000000000000000	00000000000000000000	00000000000000000000	00000000000000000000	G.711

Figura 18. Conexiones, Retardo, Jitter y Códec de 10 llamadas en curso sobre el servidor de VoIP.

Como sólo se contó con seis clientes, a partir de la séptima llamada fue necesario generar más llamadas por cada cliente.

Para el códec G711 se tuvo un comportamiento adecuado y se obtuvieron buenos resultados hasta la llamada

número 12. Con la llamada número trece el retardo y el *jitter* aumentaron drásticamente y se pudo percibir una degradación en la calidad.

En la Figura 19 se observa la degradación del *jitter* a medida que se aumentan las llamadas.

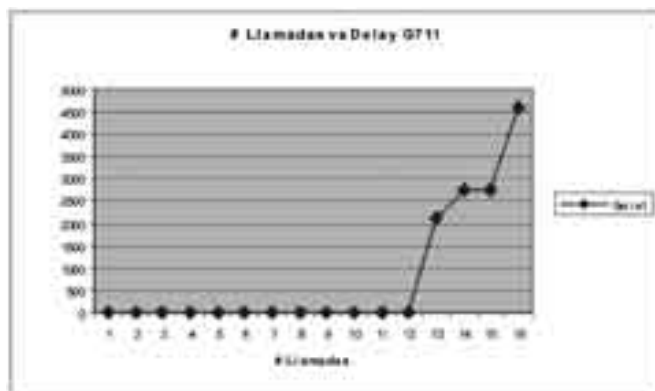


Figura 19. Observación del *Jitter* a medida que las llamadas aumentan.

En la Figura 20 se observa la degradación del retardo a medida que se aumentan las llamadas.

Como se ve en las Figuras 19 y 20 y en la Tabla IV, el *Jitter* y Retardo son

muy buenos (están dentro de los límites) hasta la llamada 12, pero en el momento en que se realiza la llamada siguiente el retardo aumenta dramáticamente a pesar de que el *jitter* se



**Figura 20.** Observación del Retardo a medida que las llamadas aumentan.

**Tabla IV.** Resumen de resultados G711.

No Llamadas	1	2	3	4	5	6	7	8
Jitter	1	1,5	1,33	2,5	2,6	2,5	2,71	4,5
Delay	3	2	2,67	3	3,8	4,67	4,57	5,13
No Llamadas	9	10	11	12	13	14	15	16
Jitter	4	3,7	4	6,75	6,62	9,36	14,26	14,87
Delay	4,5	4,5	5,64	9,17	2130	2751	2734	4588

mantiene aún apto para una comunicación adecuada de voz. Para verificar lo anterior, se puso un *softphone* con parlantes para apreciar la calidad de la comunicación y se obtuvieron las siguientes anotaciones:

Llamada 11: El sonido es de excelente calidad, no se aprecian entrecortes o chasquidos.

Llamada 12: El sonido continúa siendo excelente, aunque se alcanzan a apreciar algunos pequeños chasquidos muy esporádicos, pero nada que degrade la comunicación.

Llamada 13: La calidad se degradó apreciablemente, se aprecian fácilmente chasquidos y algunos entrecortes, toma mucho tiempo el

establecimiento de una nueva conversación.

Llamada 14: Se hace más evidente la pérdida de calidad, se aprecian muchos entrecortes y se dificulta el inicio de nuevas sesiones.

Llamada 15: No se entienden muchas partes de la conversación, se entrecorta constantemente y por largos periodos.

Llamada 16: Después de muchos intentos, se logra establecer la comunicación, pero no es entendible.

Llamada 17: Cada vez que se intentó iniciar la llamada 17, se cayó la llamada, y “tumbó” más de la mitad de las que se estaban cursando.

Es importante anotar que se repitió el experimento para verificar que la información obtenida es correcta, y que se obtuvieron los mismos datos en las dos ocasiones.

El proceso para obtener los resultados con los códigos G.729 y G.723, fue el mismo que se siguió para G.711. Los resultados obtenidos se muestran en las gráficas de las Figuras 21 y 22:



Figura 21. Observación del Jitter a medida que las llamadas aumentan con G729.

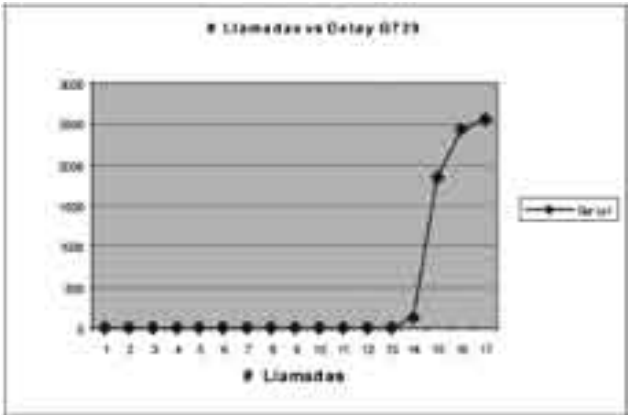


Figura 22. Observación del Retardo a medida que las llamadas aumentan con G729.

2. RESULTADOS CON G.729

Como se observa en las gráficas, el comportamiento es similar al que tienen las llamadas con G711, pero con una capacidad mayor de llamadas. Como se observa, hasta la llamada 14 el funcionamiento es estable y con buena calidad, pero en la llamada

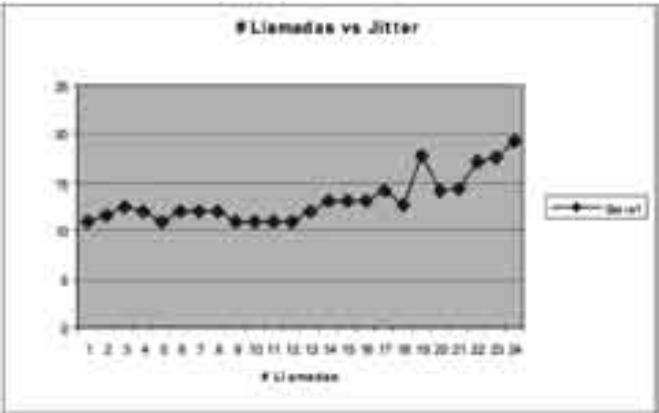
15 el retardo aumenta a tal punto, que se afecta la calidad de todas las llamadas y se empiezan a sentir entrecortes en la comunicación. A partir de este punto y hasta la llamada 17 se degrada más y más la calidad de la llamada hasta no ser entendible y/o estable. Cuando se intenta generar la

llamada 18 no se puede establecer, e incluso tumba la mayoría de las llamadas que se están cursando. Este resultado muestra que a pesar de que el ancho de banda que consume G729 es la tercera parte de G711, la capacidad de llamadas sólo aumentó en un 14%, lo que permite ver que la mayor limitante en las comunicaciones es la

contienda por el medio antes que el consumo de ancho de banda.

### 3. RESULTADOS CON G.723

Como se observa en las Figuras 23 y 24, con el códec G.723, se tiene un comportamiento similar a los dos anteriores casos, pero con una capacidad de llamadas bastante superior.



**Figura 23.** Observación del Jitter a medida que las llamadas aumentan con G723.



**Figura 24.** Observación del Retardo a medida que las llamadas aumentan con G723.

Con el códec G723 se logran cursar 21 llamadas antes que empiecen a presentarse pérdidas considerables en la calidad. A partir de la llamada

22 se empieza a degradar la calidad y en llamada 24 la comunicación es bastante difícil; al tratar de establecer la llamada 25 se desconectan muchas



de las llamadas que se estuvieran cursando.

Se deben observar dos comportamientos recurrentes en las pruebas hechas con los tres códecs:

1. El aumento de *jitter* se hace de un modo casi lineal, tiene una pendiente pequeña y no sobrepasa los límites para una buena conversación de voz, el retardo es muy pequeño y cercano a cero hasta cierto punto, pero después de éste tiene un cambio abrupto que pasa de alrededor de 5 ms a varios segundos.
2. Cuando se trató de generar nuevas llamadas en el momento en que se contaba con un retardo excesivo, se ocasionó la pérdida de conexión.

## VII. CONCLUSIONES

El desarrollo de este estudio de capacidad por métodos experimentales arroja una serie de aspectos importantes para el trabajo con VoIP en redes inalámbricas que se anotan a continuación:

Las comunicaciones de VoIP en las redes inalámbricas tienen límites abruptos, esto quiere decir que luego que se supera el límite de capacidad, una comunicación siguiente queda sin ninguna probabilidad de ser soportada por razón del retardo y pérdida de paquetes tan significativo que se presenta. Por este motivo es necesario establecer nuevos criterios de diseño con VoIP pues los tradicionalmente utilizados para Ethernet cableada no son convenientes.

Los parámetros que más incidencia tienen en el límite del número de llamadas concurrentes en un siste-

ma de VoWLAN son el retardo y la pérdida de paquetes que tiene un cambio abrupto cuando se supera una determinada cantidad de llamadas. En cambio el *jitter* presenta un comportamiento lineal, con pequeña pendiente, que se mantiene fácilmente dentro de los límites permitidos para una buena conversación.

Es muy importante en un sistema de VoWLAN poner un límite en el número de llamadas que puedan establecerse por cada punto de acceso, pues si se supera este límite se ocasionará no sólo una pérdida de calidad, sino la desconexión de las llamadas que se estén cursando a través del punto de acceso.

Debe tenerse en cuenta también el consumo de máquina que puede tenerse con la codificación de voz al usar G.729 o G.723, ya que en dispositivos móviles con poca capacidad de procesamiento puede no ser posible usar un códec diferente de G.711.

El deterioro en la calidad cuando el retardo aumenta abruptamente es fácilmente comprobable. Cada vez que se superaba el umbral de llamadas encontrado para cada códec, se escuchaba una notoria disminución de la calidad en la comunicación a través de los parlantes conectados al *softphone*, en donde se escuchaban entrecortes y múltiples ruidos.

Mientras en las redes cableadas el tipo de códec que se utiliza determina casi linealmente la capacidad (cantidad de llamadas) que puede tener el sistema, en las redes inalámbricas aunque un códec con menor consumo de ancho de banda permite un mayor número de llamadas, este efecto no es tan significativo como se notó en los

experimentos donde se comprobó que lo más crítico venía a ser la contienda por el medio.

Los resultados obtenidos de llevar a cabo las pruebas con ITG y Asterisk sirven para un proceso de validación implícita del laboratorio realizado. La única diferencia se obtuvo con el códec G.723, lo cual fue consecuencia directa de tener características diferentes para las dos aplicaciones. Pero fue muy interesante ver que tanto para G.711 como para G.729 los resultados fueron los mismos.

Con las implementaciones futuras de los sistemas de calidad de servicio, es probable que al ser la voz el servicio privilegiado, lo que se verá afectado será el desempeño en cuanto a la transferencia de datos tipo *best effort*, sobre todo si los fabricantes hacen como con 802.11 donde solamente implementaron el esquema DCF, y no se tenía un elemento central que controlara el acceso.

Sumando el conocimiento generado por estos experimentos con el conocimiento previo de los autores sobre las redes inalámbricas, se plantea la siguiente discusión: la congestión en este tipo de redes se ve que es proporcional al número de paquetes por segundo más que por el ancho de banda que consumen los paquetes (entre otras los paquetes de voz son pequeños pero requieren ser despachados inmediatamente).

Esto indica que sería interesante estudiar el impacto de diferentes tamaños de paquetes. Por ejemplo, colocar dos tramas en un paquete y por tanto bajar la tasa de paquetes a la mitad (reduciendo por tanto el *overhead* del paquete también a la

mitad) lo que sin duda reducirá la congestión y se podría esperar mejor desempeño de la red. Sin embargo, el inconveniente es que el retardo por procesamiento aumentaría y la sensibilidad a la pérdida de paquetes también, luego vendrían algunas preguntas inmediatas para resolver en trabajos futuros: ¿La disminución de paquetes perdidos debido al mejor desempeño de la red compensará el aumento en la sensibilidad a los paquetes que se puedan perder? ¿Qué es más conveniente optimizar en la práctica?

## AGRADECIMIENTOS

Los autores agradecen al Departamento de Telecomunicaciones de la Universidad del Cauca por la colaboración para el acceso a la infraestructura.

## BIBLIOGRAFÍA

- [1] IEEE Std 802.11b, IEEE Standard for Wireless LAN Medium Access Control (MAC) and Physical Layer Specifications: Higher-speed physical layer extension in the 2.4GHz band, 1999.
- [2] Wi-Fi.org. Alianza Wi-Fi. Disponible en <http://www.wi-fi.org>
- [3] Agredo G, López G. Redes Inalámbricas y Celulares como soporte a las aplicaciones de Telemedicina. Telecomunicaciones & Sociedad. Ago 2004: Volumen 2: 55 – 60.
- [4] Cisco Systems. Wireless quality-of-service deployment guide. Technical report, Cisco, 2003.
- [5] Szigeti T, Hattingh C. Quality of Service Design Overview. Disponible en <http://www>.

informit.com/articles/article.  
asp?p=357102&rl=1

- [6] Intel. Overcoming Barriers to High-Quality Voice over IP Deployments. Disponible en <http://www.intel.com/network/csp/pdf/8539.pdf>
- [7] J. Chou. "Design a successful VoWLAN system". Wireless Net DesignLine. Sep. 2005. Disponible en <http://www.wirelessnetdesignline.com/how-to/170101775>
- [8] F. Mlinarsky. "Metrics and Methods Bring VoWLAN Success". Wireless Systms Design. Mar. 2005. Disponible en <http://www.wsdmag.com/Articles/ArticleID/10003/10003.html>.
- [9] Distributed Internet Traffic Generator. Università degli Studi di Napoli "Federico II" (Italia) Disponible en <http://www.grid.unina.it/software/ITG/>
- [10] Asterisk "The Open Source PBX" Disponible en <http://www.asterisk.org/>

## AUTORES

### Guefry Leider Agredo Méndez.

Docente de la facultad de Ingeniería Electrónica y de Telecomunicaciones de la Universidad del Cauca. Estudiante de la maestría en Electrónica y Telecomunicaciones de la Universidad del Cauca. Investigador del Grupo de I+D en Nuevas Tecnologías en Telecomunicaciones (GNTT). Áreas de interés: Comunicaciones Inalámbricas, Voz sobre IP, Servicios de Internet. e-mail: [gagredo@unicauca.edu.co](mailto:gagredo@unicauca.edu.co)

### Jaime Andrés Gaviria Molano.

Ingeniero Jefe de Servidores y Servicios de Internet de la Universidad del Cauca. Estudiante de la maestría en Electrónica y Telecomunicaciones de la Universidad del Cauca. Áreas de interés: Voz sobre IP, Reconocimiento de Voz. e-mail: [jgaviria@unicauca.edu.co](mailto:jgaviria@unicauca.edu.co) 